جامعة المرقب / الخمس

كلية التربية - سوق الخميس

اساسيات البرمجة بلغة الجافا

الدوال في لغة الجافا

Java Methods

المحاضر : سالم مسعود الدروقي

Out lines

- ماهي الدوال وما هي فوائدها (What's Methods & Why Methods)
- انواع الدوال في لغة الجافا (Programmer –defined - Built-in)
- تصنيف الدوال المعرفة من قبل المبرمج .
- الاعلان عن الدوال (الشكل العام للدالة Method Structure)
- استدعاء الدوال وتمرير المعاملات للدالة Method invocation & Passing Parameters
- الدوال والمصفوفات (ذات البعد الواحد / ذات البعدين) Arrays & Methods
- التحميل الزائد للدوال Method Overlooding

ما هي الدوال وما هي فوائدها
What is Methods & Why Methods

- الدالة هي عبارة عن مجموعة من التعليمات المجمعة تحت اسم واحد.
- يمكن استدعائها في اي نقطة في البرنامج من (خلال كتابة اسمها في البرنامج).
- وبعد تنفيذها يتم الرجوع الى نقطة الاستدعاء في الدالة الرئيسية.
- تساعد الدوال في تنظيم وتنسيق هيكلية البرنامج.
- تقسيم البرنامج الى مجموعة برامج فرعية (دوال) بحيث يكون لكل منها وظيفة محددة.
- استخدام الدوال يسهل عملية متابعة وصيانة الشفرة المصدرية للبرنامج.
- تساعد في التقليل من تكرار كتابة الاكواد.

(Types of Methods in Java) انواع الدوال في لغة الجافا

- **تنقسم الدوال في لغة الجافا الى نوعان رئيسيان هما :**
- **الدوال الجاهزة Build-in:** وهي مجموعة الدوال المبرمجة مسبقا ضمن حزمة المترجم الخاص بلغة الجافا مثل (الدوال الرياضية ، دوال التعامل مع النصوص ، الدوال العامة)
- **الدوال المعرفة من قبل المبرمج User-defined:** وهي مجموعة الدوال التي يتم انشائها من قبل المبرمج لاداء وظيفة معينة وتتكون جزئين رئيسيين هما :
- راس الدالة (Method Header)
- جسم الدالة (Method Body)

```
<Method Header>()
{
<Method Body >
}
```

معايير تصنيف الدوال المعرفة من قبل المبرمج

- **قابلية الوصول Access Modifier :**
- **عامة Public:** يمكن الوصول اليها من كافة الفئات في المشروع.
- **خاصة Private:** لا يمكن الوصول اليها الا من داخل الفئة المعرفة فيها.
- **محمية Protected:** لا يمكن الوصول اليها الا من خلال من الفئة المعرفة فيها الفئات الموروثه منها

معايير تصنيف الدوال المعرفة من قبل المبرمج

- **المشاركة بين الكائنات (Static , non static):**
- الدوال يتم تعريفها داخل الفئات و التي يمكن ان نشتق منها مجموعة من الكائنات وفي هذه الحالة توجد هنالك نوعين من الدوال:
- **غير مشتركة Non Static (instance member):** اي انه لكل كائن مشتق من الفئة قيما خاصة لكافة متغيرات الدالة وفي مواقع مختلفة من الذاكره ولا ستدعاء هذه الدالة يجب اولا اشتقاق كائن(object) من الفئة المحتويه على الدالة.
- **مشتركة Static (Class member) :** اي ان هذه الدالة مشتركة (لها موقع واحدة في الذاكرة) بين كافة الكائنات المشتقة من الفئة المحتوية على الدالة وعند استعاء هذا النوع من الدوال لانحتاح الى اشتقاق كائن من الفئة المحتوية على الدالة.

معايير تصنيف الدوال المعرفة من قبل المبرمج

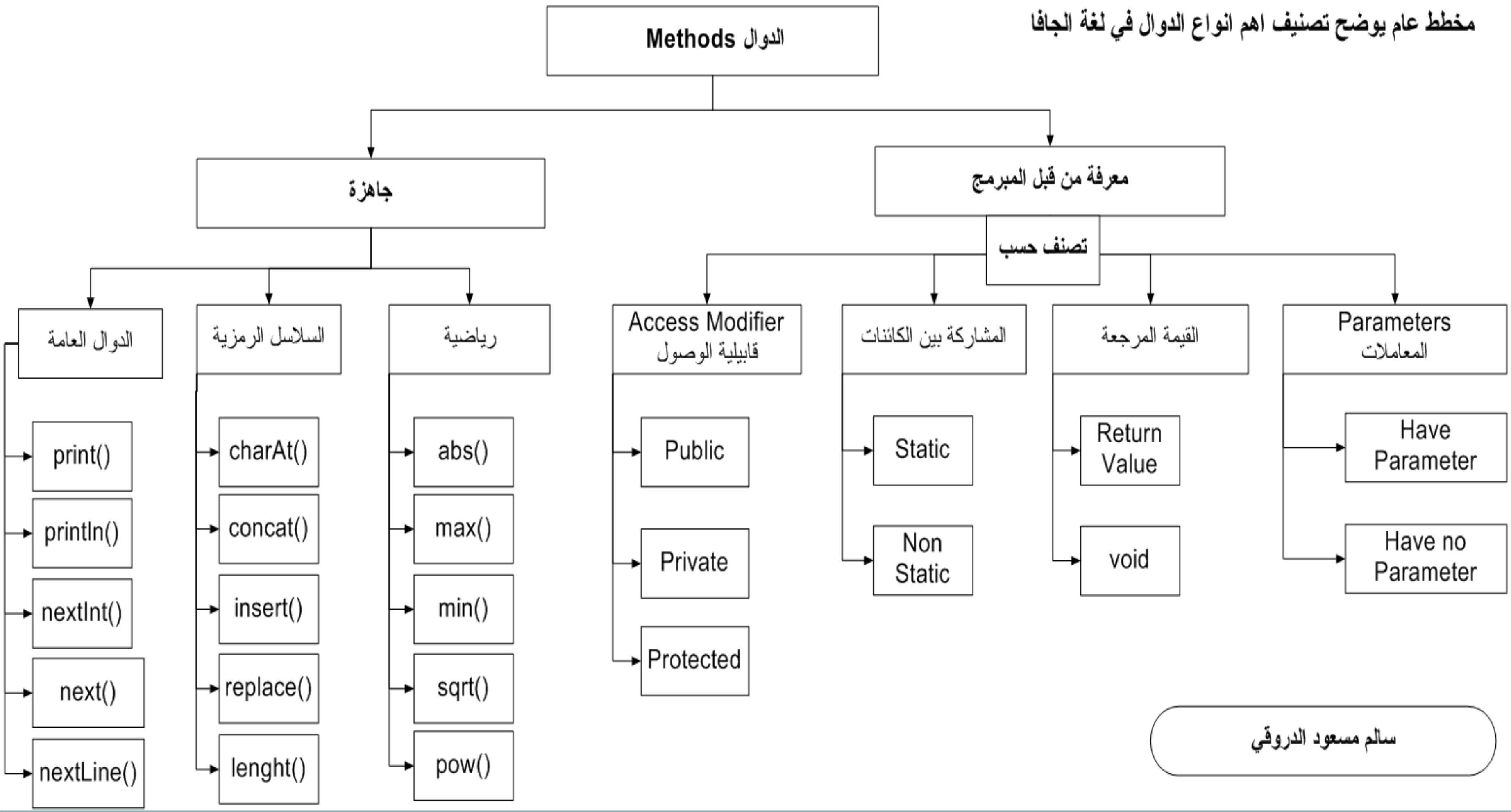
- **القيمة المرجعة لسطر الاستدعاء Returned Value to calling code:**
- **دوال ترجع قيمة (Getter)Return Value :** هذا النوع من الدوال يقوم بتنفيذ تعليمات محددة ثم يقوم بارجاع قيمة (يتم تحديد نوعها اثناء تعريف الدالة) الى سطر الاستدعاء بعد انتهاء التنفيذ ويجب ان يحتوي جسم الدالة علىالكلمة المحجوزة return .
- **دوال لا ترجع قيمة (Setter)Return no Value(Void):** هذا النوع من الدوال تقوم بتنفيذ تعليمات محددة دون ان تقوم بارجاع قيمة الى سطر الاستدعاء (Calling code) بعد انتهاء التنفيذ.
- سطر الاستدعاء Calling Code: هو السطر الذي تم عندة استدعاء الدالة في الدالة الرئيسية (Main Method)

معايير تصنيف الدوال المعرفة من قبل المبرمج

- احتوئها على المعملات **With or without Parameters:**
- **دوال لا تحتاج الى تمرير معاملات Have no Parameter:** وهي دوال لا تحتاج الى تمرير قيم اثناء استدعائها حيث لا يتم كتابة اي قيم بين قوسي الدالة.
- **دوال تحتاج تمرير الى معاملات Have Parameter:** وهي دوال تحتاج الى تمرير (ارسال) معاملات (توابت او متغيرات) اثناء استدعائها.
- يتم كتابة المعاملات كقيم ثابثة او متغيرات تحمل قيماً بين قوسين امام اسم الدالة اثناء استدعاء الدالة.

مخطط يوضح تصنيف الدوال في لغة الجافا

مخطط عام يوضح تصنيف اهم انواع الدوال في لغة الجافا



سالم مسعود الدروقي

الشكل العام للاعلان عن الدوال

- visibility [static] return-type method-name (parameter-list)

```
{
 statements
...
 }
```

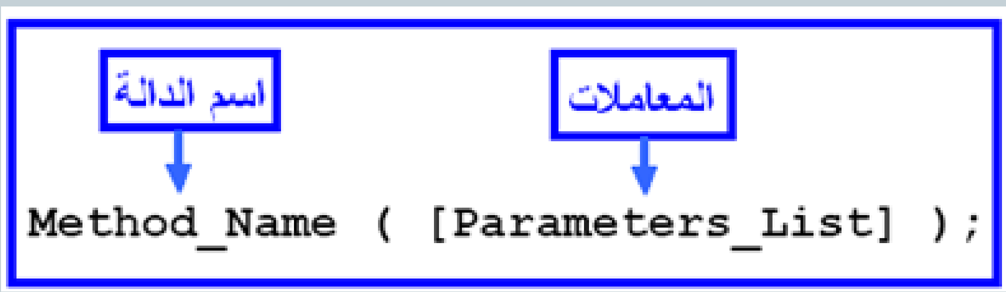
مثال لتعريف دالة عامة ومشتركة بين الكائنات ترجع قيمة من النوع الصحيح وتحتاج الى تمرير ثلاث

معاملات من النوع الصحيح

```
Public static int maxm(int num , int num2 , int num3)
{
......
.......
int z
return z
}
```

استدعاء الدوال وتمرير المعاملات

- يتم استدعاء الدالة عن طريق كتابة اسمها في الدالة الرئيسية متبوعا بقوسين يتم بينهما كتابة قيم المعاملات المطلوب تمريرها الى الدالة إن وجدت.



- مثال لا ستدعاء الدالة في المثال السابق :

- maxm(5,7,3);

طرق تمرير المعاملات للدالة

- المعاملات هي عبارة عن قيم (متغيرات او ثوابت) يتم تمريرها الى الدالة اثناء استدعائها من خلال كتابة قيم او متغيرات مناظرة للمتغيرات المعرفة في راس الدالة في جملة الاستدعاء.
- تسمي المعاملات الموجودة في الدالة الرئيسية (المكتوبة امام جملة الاستدعاء) بالمعاملات الفعلية بينما تسمى المعاملات المناظرة لها (المكتوبة في راس الدالة) في الدالة المستدعاة بالمعاملات الصورية او الشكلية.
- يجب ان تتطابق المعاملات الفعلية والمعاملات الصورية من حيث انواعها وعددها وترتيبها.

تمرير المعاملات بالقيمة Pass-By-Value

- في هذا الطريقة يتم ارسال نسخة من المعامل الفعلى الى المعامل الصوري.
- كلا المعاملين (الفعلي و الشكلى) لا يشتركان في موقع و احد في الذاكرة.
- المعامل الفعلي لا يتأثر بالتغيرات التي تحدث على المعامل الصوري.
- هذا النوع من تمرير المعاملات يتم تطبيقه تلقائيا عندما يكون نوع المعاملات الفعلية من انواع البيانات البدائية (Primitive Data Types) مثل int,double,float.

تمرير المعاملات بالعنوان (المرجع) (بالاشارة) Pass-By-Reference:

- فيها يتم ارسال عنوان المعامل الفعلى في الذاكره الى المعامل الصوري المناظر له في الدالة المستدعاة.
- في هذه الحالة المعامل الفعلي و الشكلي يشيران الى نفس موقع الذاكرة (يشتركان في موقع الذاكرة)
- اي تغير يحدث في قيمة المعامل الشكلى سيتم تطبيقه على المعامل الفعلى في الدالة الرئيسية.
- هذا النوع من تمرير المعاملات يتم تطبيقه تلقائيا عندما تكون المعاملات الفعلية من نوع كائن (Object) مثل المصوفات.

الدوال و المصفوفات
Methods & Arrays

- الشكل العام لتعريف دالة تستقبل مصفوفة من البارمترات ولا تعيد قيمة

```
Access_modifier   returen_type   method_name (array_type    array_name[])
{
}
```

- الشكل العام لتعريف دالة تستقبل مصفوفة من البارمترات وترجع مصفوفة من القيم

```
Access_modifier  returen_type[]   method_name (array_type    array_name[])
{
}
```

التحميل الزائد للدوال
Methods Overloading

- هو عبارة عن كتابة اكثر من دالة تحمل نفس الاسم في فئة واحدة مع اختلاف توقيع كل دالة (Method signature) لكي يتم التميز بينها.
- توقيع الدالة Method signature : يتمثل توقيع الدالة في قائمة البارمترات الموجودة بين قوسي الدالة ولكي نستطيع تعريف اكثر من دالة بنفس الاسم داخل فئة واحدة فان هذه الدوال المعرفة يجب ان تختلف في احدى ثلاثة اشياء :
- عدد المعلاملات Number of parameters
- نوع المعلاملات Data type of parameters
- ترتيب نوع المعاملات Sequence of Data type of parameters

امثلة عن التحميل الزائد للدالة

دالتين مختلفتين في عدد المعاملات

```
public static int max(int x , int y)
   {
     // Method body
   }
```

```
public static int max(int x , int y,int z)
   {
     // Method body
   }
```

دالتين مختلفتين في نوع المعاملات

```
public static int max(int x , int y)
   {
     // Method body
   }
```

```
public static int max(double x , double y)
   {
     // Method body
   }
```

دالتين مختلفتين في ترتيب نوع المعاملات

```
public static int max(int x , double y)
   {
     // Method body
   }
```

```
public static int max(double x , int y)
   {
     // Method body
   }
```

اكتب برنامج يقوم باستدعاء دالة تقوم بحساب مساحة مثلث علما بان

قاعدة المثلث =تساوي 4.5 متر وارتفاعه 6.6 متر على ان يتم طباعة المساحة في الدالة.


```
public static void main(String[] args)
  {
   double B=4.5,H=6.6;
   triangle_area(B,H);
  }
  public static void triangle_area(double x,double y)
  {
     double area;
     area=(0.5*x)*y;
   System.out.println("Area="+area );
  }
```

نفس المثال السابق على ان تتم الطباعة في الدالة الرئيسية


```
public class JavaApplication1 {
    public  static  void  main(String[] args)
  {
    double  B=4.5;  H=6.6;   double  area;
    area= triangle_area(B,H);
   System.out.println("Area="+area );
  }
  public  static  double   triangle_area(double  x,double  y)
  {
   return (0.5*x)*y;
  }  }
```

اكتب برنامج يقوم بتخزين قيم لعناصر مصفوفة ثم يقوم باستدعاء دالة تقوم بطباعة هذه القيم.


```
public static void main(String[] args)
  {
    int m[]= new int[5];
    m[0]=5; m[1]=15; m[2]=55; m[3]=3; m[4]=7;
    print_arr(m);
  }
  public static void print_arr(int arr[])
  {
     for (int i =0 ;i <5;i++)
        System.out.print(arr[i]+" ");
  }
}
```

اكتب برنامج يقوم باستقبال مصفوفة دات بعدين مكونة من 10 عناصر 2*5 من النوع الصحيح ثم يقوم باستدعاء دالة تقوم بحساب اكبر قيمة في المصفوفة وكذلك دالة لحساب اصغر قيمة.

- الدالة الرئيسية

```
public static void main(String[] args)
   {
     Scanner input = new Scanner(System.in);
    int  m[][]= new int[2][5];
     for (int i =0 ;i <2;i++)
       for (int j =0 ;j <5;j++)
        m[i][j]=input.nextInt();

    int max_element= max(m);
    int min_element= min(m);
    System.out.println("max="+max_element);
      System.out.println("min="+min_element);
   }
```

تابع المثال في الشريحة السابقة

- دالة حساب اكبر قيمة

```
public static int max(int arr1[][])
  {
     int max=arr1[0][0];
     for (int i =0 ;i <2;i++)
       for (int j =0 ;j <5;j++)
       if (arr1[i][j]>max)
          max=arr1[i][j];
     return max;
  }
```

تابع المثال في الشريحة السابقة

- دالة حساب اقل قيمة

```
public static int min(int arr2[][])
  {
      int min=arr2[0][0];
      for (int i =0 ;i <2;i++)
         for (int j =0 ;j <5;j++)
        if (arr2[i][j]<min)
            min=arr2[i][j];
      return min;
  }
```

مثال :

النهاية

شكر لحسن استماعكم

المحاضر : سالم مسعود الدروقي

Salemaldrugi@gmail.com
Salemaldrugi@yahoo.com